

# SINC - LINK

Vol. 4 No. 1

Jan. - Feb. 1986

SINC LINK IS A PUBLICATION OF THE TORONTO TIMEX-SINCLAIR USERS CLUB AND IS ISSUED 6 TIMES A YEAR. COPIES OF THE NEWSLETTER ARE \$1.50 EACH FOR NON-MEMBERS. CLUB MEMBERS RECEIVE A FREE COPY AS PART OF THE \$20.00 ANNUAL MEMBERSHIP FEE.

NEWSLETTERS ARE EXCHANGED, FREE OF CHARGE, WITH OTHER TIMEX-SINCLAIR USERS GROUPS

ALL MATERIAL IS IN THE PUBLIC DOMAIN AND CAN BE REPRINTED, PLEASE CREDIT THIS PUBLICATION AND THE AUTHOR IF YOU COPY MATERIAL.

SEND CORRESPONDANCE TO:

SINC LINK  
C/O JOHN ROACH  
88 DUNDALK DRIVE, UNIT 23  
SCARBOROUGH, ONTARIO  
M1P 4X6  
(416)-297-6011

## NEWSLETTER INDEX

PRESIDENT'S MESSAGE	P. 2
LARKEN DD LOADER	P. 2
SINCBITS	P. 3
ZX81 NEWS	P. 4,5
SPECTRUM TAPE BACK-UP	P. 6
QL CORNER	P. 7
BOB'S NOTEBOOK	P. 8
LUNAR CALCULATING	P. 9
TS2068 PLOT & SCROLL	P.10
HINTS-N-TIPS	P.11
DISK DROPPINGS	P.12

## THE "BUZZ"

What is that annoying Buzz in the 2068? It always seems to happen when typing in a lengthy BASIC program, doesn't it? Well, there are good reasons for it because your machine is trying to tell you that you goofed somewhere. If you have never heard the Buzz then try the following:

CLEAR 26790 - (for THE TS2068 ROM)  
CLEAR 23850 - (for the SPECTRUM ROM)

Now type anything and there it is....that annoying Buzz. The computer is telling you that the Basic Memory is too full. You lowered RAMTOP so that there are only a few bytes for Basic and it cannot accept them. If there was a little more room in memory you may have seen the error messages:  
" MEMORY FULL" or " NO ROOM FOR LINE"

Another situation where Buzz occurs is when you type in a line longer than the 23 screen lines. You may or may not want to try this one. However, any more typing than 23 lines is not ignored even though it goes off the screen and you can't see it. The Buzz will sound for each key hit just to annoy you and hint that you have typed a long, long line. Getting back to typing in a lengthy Basic program; you eventually get tired and accidentally confuse your shifts on the top row of keys.

This causes a control character to be entered into a line where it shouldn't be (often changes colour of listing). By re-LISTing the program the length of that line may get corrupted and makes your 2068 think it's too long. This is where it tells you to Buzz off. Only extensive PEEKing and POKEing can repair the damage in the program.

from: SPECTRUM Users Manual, Chapter 24  
compiled by Fred Schakel  
London T/S Users Group

**TORONTO TIMEX - SINCLAIR USERS CLUB**

P. O. Box 7274 Stn. A Toronto, Ont., M5W 1X9  
Canada

## LARKEN DISK DRIVE OPERATION by George Chambers

One irritating thing about the LARKEN drive is the need for some 26 keystrokes to get the drive operational. This article provides an effective solution to the problem. What I have done is modify an ordinary tape cassette to do the task for me.

The modification consists of removing the existing reels in the cassette body and replacing them with a 12 inch continuous loop of tape. This length of tape is sufficient to hold a short "boot" program which gets the drive up, and to load a program called "menu.B1".

What we are going to do is to remove the reels and tape from the cassette, and mount two additional rollers in the cassette body to guide the tape loop. Then make up an endless loop of tape, install it, and close up the cassette.

Now to get down to details. First pick a cassette which is fastened with screws. This will make it easy to take apart. Actually you will need a second cassette because we need a second pair of the little rollers and their bearing pins, to mount inside the first cassette.

Open both cassettes. Discard the reels, lubricating paper inserts, and tape. (Save a few feet of tape to make up the tape loop). Carefully remove and save the two rollers and pins from one of the cassettes. The two additional rollers are to be mounted inside the other cassette near the two knockout tab recess. The idea is to provide as long a tape path as possible.

You will need a small drill just about the diameter of the roller pins. Assemble the cassette case and drill two small holes completely through the cassette, just where the new rollers are to be installed. These holes should provide a press fit for the roller bearing pins. Make sure the holes are perfectly at right angles to the cassette body, so that the rollers roll freely, and do not bind.

Open the case and install the pins, and rollers. See that they turn readily.

Now measure a length of recording tape. It should be just the length of the path around the four rollers in the cassette. Splice the tape. You may have to make several tries. If the tape loop is too short it obviously will not go over the rollers; if it does it may still be too tight and bind. On the other hand it may be too long. If it is too long this will be apparent by a bulge of tape appearing near the pressure roller in the tape deck when it is in operation, instead of being drawn promptly into the cassette.

Once you have your cassette made up it will be necessary to save a program onto it. A suitable program is shown elsewhere in this article. The program boots the drive and calls up a program. I make use of a small Basic program on each disk which is either a MENU program, or a simple loader for a main program such as TASWORD.

Type the program in, then start recording on this tape. Simply SAVE the program by entering SAVE "boot" LINE 10. Watch the SAVE process, and stop the recorder IMMEDIATELY the SAVE ends. Test by reloading, or better still MERGE it. (This way you will not lose the program you have typed in.) Chances are that you have SAVED it over the splice, in which case you will get a load error. Record it again, and again if necessary, until you have a good save. Remember it is critical to stop the recorder immediately after the save. No dallying here.

boot program:

```
10 CLEAR 61000: OUT 84,64: PRINT USR 63488
20 REM load "menu.B1"
```

## PRESIDENT'S MESSAGE

There has been a development in our club newsletter which has been very gratifying to the newsletter editor and myself, and which is worthy of mention this issue.

This has been the appearance of regular contributors to our newsletter. First there was Ian Robertson with his "SINCBITS". Then Peter McMullin with his "ZX81 NEWS AND RESOURCES", and John Burns and Dave Ridge with a continuing column "TELECOMPUTING".

May I extend a warm thank you on behalf of our membership to last years columnists, who made our newsletter such an outstanding success. We look forward to the continuance of their columns in the coming year.

This year we are delighted to welcome three more columnists to our newsletter. We are to have Peter Hacksel, who will be covering the Disk Drive scene, specifically the LARKEN Drive. Also Enore Gardinio with a regular column devoted to the new QL; and not least, Karim Rahamet with a Hacker's column.

At the same time, we must applaud the work of our editor, John Roach, without whose efforts this whole newsletter production would not have been possible. Thanks, John.

Yours in computing, George Chambers.

# SINCBITS

Ian Robertson  
Compuserve 72167,3401  
FIDO NET 148/MODE 608

UPDATES: Those of you who correspond via FIDONET - PLEASE NOTE MY NEW NET 148 NUMBER. It's nice to see more and more people stating in print that the Timex/Sinclair market has not died, but instead has developed into a marketplace with fewer products, but of a much higher quality. There is another 2068/Spectrum compatible DDI on the market, it is the John Oliger Disk Drive Interface. It is available in four configurations, from JOHN OLIGER COMPANY, 11601 WHIDBEY DRIVE, CUMBERLAND, IN 46229; a) the two bare boards \$44.00 b) as a kit with all parts \$98.00, c) assembled and tested (without NMI) \$120.00, and lastly, d) assembled and tested (with NMI) for \$130.00. NMI stands for "non maskable interrupt", which simply means that ram contents can be dumped to disk at the touch of a button. All prices are in US funds and include the JLD SAFE DOS on 8k eprom plus postage. In order to operate this DDI you must also buy the Oliger expansion board. I now have my bare boards assembled and am in the process of testing. On power-up the JLD SAFE DOS checks to see which ROM is being used - I can report that this aspect really works. One last word "this DOS IS FAST", more in the next issue. I received a replacement microdrive back from Sinclair Research, in about four weeks time, and at no cost! Not too shabby I must say! The only problem is - this one is also noisy and slightly out of alignment!! At least it is better than the one I returned. Now that the cost of 256k chips is coming down, one smart UK company will install them inside the QL, to enable the owner to run 512K. The only side effect is that the system apparently does not run as fast as a similar QL with outboard RAM. QL owners take note!! My wife would like somebody to keep reminding me that the present version of the QL will probably be replaced by either QL II or a completely different business type machine and further that I already have too many obsolete computers.

TS2068: There is definately a mouse in my house! It is a Radio Shack CoCo2 mouse connected to the Zebra Graphics Pad Interface (with a slight modification). AND IT WORKS! To be able to use Tech Draw without all that "screen spatter" is quite a treat. Now that Anchor has dumped their TS2050 modem line, there are a lot of bargains around. Zebra Systems Inc., 78-06 Jamaica Ave., Woodhaven NY 11421, telephone (718) 296-2385 are selling the complete modem for \$72.95 US (including shipping). Various others are selling the board/cables for \$25.00. Zebra are also selling an untested/uncased board for \$13.00 each, \$28.00 for three and \$73.00 for a quantity of ten. The message is - IF YOU WANT A 2050 MODEM GET IT NOW. The latest version of MSCRIPT is available from Zebra. Looks good. An even more up-to-date version is available from Jack Dohany, 325 O'Connor Street, Menlo Park, CA 94025. He sells upgrades ONLY to bona fide MSCRIPT owners - pirates stay away. He operates his own company called "Fairware" and needless to say the prices are "fair".

SPECTRUM: My big news item for the Spectrum is that I now have the Technology Research "Beta" DDI up and running on my "plus" complete with IF1 and two microdrives. WHAT A SYSTEM!!! It

comes with a 5.25" utility disc, which I can use on the quad drive or can be transferred to a 3" disk. Very flexible! It is possible to switch from DOS to Sinclair Basic by software commands. This means that disks, microdrives and cassette can be used at any time without a problem. Using the "magic button" on the DDI I can dump the RAM contents to disk at any time. This means that ALL games can be put on disk and can come up running at the location on screen when the button was pressed. So far I have about 40 games on disk - and they take a "slow" nine to ten seconds to load. Even for a non-games person like myself this makes playing games easier (and tempting). The 3" disks format to 160k and this allows 3 games to be saved to each side of the disk. The Abbeydale SPDOS formats to 184k on each side of the disk (as in the Ramex DDI and probably the KEMPSTON). The AMX mouse package is now \$70.00 (pounds sterling), down from the original \$80.00. Hope it goes even lower. There has been considerable press regarding "Lenslok", a software protection device which looks like a small folding magnifying glass. The observations have been mixed - the software people like it and it appears that the users are not quite so sure. Think about it, how would you like to look at a "very bright" screen through a magnifying lens to load a stubborn program. Apparently FIREBIRD used it on their latest program "Elite" and are quoted as saying that they are considering not using it in the future, due to very bad reviews about their instructions for using lenslok. If you are into computer games, then you MUST try "Tomahawk" by Digital Integration. It seems to be the best flight simulator available, even if it is a helicopter (but what a helicopter!!).

DISK DRIVES: I will comment only on the financial aspect of using DD with either the Spectrum or the 2068. The following approximate prices are in our lowly Cdn. dollar, with customs duty and delivery included. The prices will also include 2 drives, power supply and cases except where noted as DDI (disk drive interface) only. RAMEX/ANDEK III system \$520.00 from RAMEX INTERNATIONAL, 17620-26 MILE ROAD, WASHINGTON, MI 48094 telephone 313-781-5800. ZEBRA/TIMEX \$810.00, AERCO \$825.00 FROM ACME ELECTRIC ROBOT CO. BOX 18093, AUSTIN, TEXAS 78760 telephone 512-451-5874. JOHN OLIGER (DDI) \$270.00 (includes JLD extender board at \$84.00 Cdn), this price can be reduced if you do-it-yourself (see UPDATES above). LARKEN (DDI) \$135.00 from LARKEN ELECTRONICS, RR #2, NAVAN, ONTARIO K4B 1H9. TECHNOLOGY RESEARCH "BETA" from Bob Dyl of ENGLISH MICRO CONNECTION, 15 KILBURN COURT, NEWPORT RI 02840, telephone 401-849-3805, Spectrum only DDI \$290.00. Bob also sells the Kempston (SPDOS) DDI - if anybody has one I would appreciate a comment or two. As you can see the world of mass/fast storage is an expensive one. Remember one thing - unless you want to use a DD system exclusively for either games or business applications "almost any system will do", therefore, before you buy "ask somebody who has one".

TS1000/ZX81: Anyone interested in joining a COMPUSA USER'S GROUP should write to the following address; CUSS, P.O. Box 2186 Inglewood, CA 90305. It is being formed by Tony Gomez and Ed Grey. If you want info faster Tony can be reached at FIDO Net 102 Node 602 and Ed can be reached at FIDO Net 102 Node 106 and on CompuServe 75236, 3233. By the way, CUSS stands for "Compusa Users Support Group". I have a copy of their user questionnaire, if anyone is interested.



Welcome to our first issue for 1986. This column is, once again, devoted to a lengthy product review. I'm planning a catch-up on resource/supplier listings next issue, but a couple of items deserve immediate mention:

**>>> HARDWARE HARDBOYIST HANDBOOK! <<<<**

ZX81 hardware expert Bruce Taylor, of Budget Robotics & Computing, Box 18616, Tucson, AZ 85731, has announced the release of his book, **BUILD A MICROCOMPUTER- CONTROLLED ROBOT AND OTHER COMPUTER CONTROL PROJECTS**. (quite a mouthful!)

TAB Books had owned the rights for over a year, then decided not to publish. After this runaround, Mr. Taylor's company bought back the rights and is now publishing the book. This book presents the design of an award-winning robot in thoroughly documented project form, and promises to be a valuable reference on interfacing just about anything. Printed circuit boards will be available from Budget Robotics for every project in the book. Chapter topics include Adjustable Power Supply, 8255 PIO Board, Digitalker voice synthy, Stepper Motor Drivers, Optical Encoders, Polaroid Ultrasonic Ranging kit, and others. 200+ 8.5"x11" pages, 80+ illustrations, 60+ tables, loosebound. Price: U.S.\$13.00, ppd.

For a sample of previously published work, see "Robotics on a Budget", p.18, SYNC Vol.3 No.4; also "Home Control on a Budget", p.31, SYNC Vol.4 No.1; both by Bruce C. Taylor.

Budget Robotics holds rights to an improved version of the Computer Continuum buffered bus expansion board, as well as the Zodex RX81, an inexpensive general purpose parallel I/O board. Write for details- Bruce replies promptly.

**>>>>> HOLY MODEM, BATHAN! <<<<<<**

Westridge 2050 modems, tested, with phone cables but no case nor power supply are available from DAVE CLIFFORD, 13910 HALDALE AVE., GARDENA, CA 90249, (213)-516-6648. The price? U.S.\$25.00 for one, three or more for \$22.50 each, ten or more for \$20.00. That's U.S. bucks, including delivery. At time of writing, Dave indicated he had about 1500 left. However, he also has an undisclosed number of untested boards which are missing a couple of minor parts (an LED & resistor). Plans are to sell these as-is at an even cheaper price, including a complete schematic & parts ID for the hardware hacker.

Needless to say, I ordered my modem the same day I heard about Mr. Clifford, as well as a copy of MINI-XMOD 1.7, at U.S.\$20.00 from WEYNIL CORP., Box 5904, Bellingham, WA 98227-5904.

MINI-XMOD is terminal software which is said to be the best available for the ZX81/TS1000 (or the 2068!). Dave Clifford indicated that he would be working on a version of MINI-XMOD for the 2068, first a 32-column version, then a 64 col. version. It would be an interesting challenge to work up a 64 column version for the JLD Video TS1000! For a short but sweet review of MINI-XMOD 1.7, see Telecom Treats, in Sinc-link Vol.3 No.6 (last issue), by Mel Richardson.

**SIDENOTE:** The "warped checkerboard" graphic on the bottom of page 2, last issue, was created on a TS1000 using a graphics utility from Callisto Software, transferred to Sinc Artist for enhancement, then printed out size-as on this Epson MX80.

**>>> HARDWARE USER REPORT <<<<**

**SUBJECT:** The AERCO FD-ZX Floppy Disk System.

**SOURCE:** AERCO, 7606 ROBALO RD., AUSTIN, TX 78757

FD-ZX Floppy Disk Controller Board & DOS: U.S.\$179.00

Auto Boot ROM Board: U.S.\$ 59.00

Auto Boot ROM Board with Centronics I/F: U.S.\$ 99.00

Complete packaged systems are also available.

SEE "Bug Alert", page 4 last issue, for a discussion of the Boot ROM / Centronics Board.

**BASIC TECHNICAL SPECIFICATIONS:**

**Drive type:** Shugart compatible 3",5" or 8" drives,  
1 or 2 side, single or double density,  
35 or 40 tracks/side.

**# of Drives:** up to 4

**DS/DD Capacity:** 400K bytes

**Disk Format:** IBM readable, 10 sectors  
per track, 512 bytes/sector.

**# Files per Disk:** up to 20 (as 4-track "pages")  
up to 6 (as 12-track "pages")

**Memory Use:** 2K DOS in EPROM, and FDC at 12-14K  
FDC I/O is memory mapped.

**Format DSDD:** approx. 35 sec.

**Copy DSDD-DSDD:** approx. 1 min. 5 sec.

**Avg. LOAD/SAVE:** 3 sec. (20k page)  
5 sec. (60K page)

**Error Reports:** WRITE PROTECTED; INSUFF. RAM; DISK ERROR

**Physical size:** Circuit board is approx. 4.5"x6",  
plus gold finger extensions.

**Special ICs:** 1797 FDC (Siemens); 9216 Data Separator

WHAT DO YOU SEE when you open the box containing your newly-arrived AERCO FD-ZX? One uncased controller board, a computer bus ribbon cable, one floppy disk, and a 16 page manual. What else do you need? Disk drive(s), drive cables, cases, & power supply. Aerco's manual gives reasonably easy-to-follow instructions for hooking up your system, and has several useful illustrations. The power supply must provide +12V at 2-3 amps, and +5V at 3 amps. The drive cable required is a standard IBM/Shugart 34 conductor pc-edge type. If you follow Aerco's instructions to the letter, you will likely void any warranty on your new drives. The FDC Board gets its +5V and +12V supplies via pins 2 and 34 on the disk drive edge connector. Pin 34 is normally unused, but pin 2 is connected on some drives. Aerco recommends you cut any traces going to pins 2 or 34 on the drive, and jumper these conductors over to the power supply pins. This way, you connect the power supply only to one drive, and the FDC Board & other drives get powered thru the ribbon cable. The result is tidy, but I hate having to hack up a new drive! The alternative would be to cut wires 2 & 34 adjacent to the controller card, and run separate power leads to everything. I did it Aerco's way.

Having hooked up drives & power supply, the FD-ZX plugs into the supplied ribbon cable; then the RAM pack, etc. onto the ribbon cable's male edge extender, and we're ready to see what's on Aerco's "master" disk.

On application of power, the K cursor appears. Following Aerco's instructions, the master disk is placed in Drive A. The command RAND USR 13303 initialises the DOS, and loads "page 1" from the disk. The following menu appears on the screen in about 3 seconds:

WELCOME TO SADOS+ 16K 8/83

```
C COPY DISK (REQUIRES 2 DRIVES)
D DISK UTILITIES
E ENTER NAMES IN DIRECTORY
F FORMAT DISK IN DRIVE A
H HEX MONITOR
L LOAD MEMORY FROM DISK
Q QUIT
S SAVE MEMORY ON DISK
```

Nothing too surprising here. Some further exploration of the manual reveals several interesting facts. All DOS functions are accessed via RAND USR commands. The "DOS" program supplied on page 1 of the master disk serves two purposes: 1) It is a "user-friendly interface" (oh cliché, cliché!) to the DOS USR routines, and 2) It contains the Disk Directory.

Two different DOS control programs are supplied, a "16K" version, and a "64K" version. (All capacities specified here are for a 40 track, DSDD drive.)

The 16K DOS permits 20 programs or data files to be saved on a disk. The disk is divided into 20 4-track pages. Any program saved occupies 1 page, regardless of whether it is a 1K program or a full 16K program. Each "16K" disk page will actually hold up to 20K of data. Thus it is extremely unlikely that you will ever use the full potential storage capacity of the disk. The 64K DOS assigns 6 pages per disk, each page having a capacity of 60K bytes. It is otherwise identical to the 16K DOS program.

The Aerco DOS does not record a filename with each program, nor does it automatically maintain a directory. Programs are loaded & saved by page number. It is the user's responsibility to keep the directory on page 1 updated.

Converting a breakable program to a disk version is quite easy. Aerco uses sequential USR calls for SAVE & LOAD commands: RAND USR (12720+PAGE) to SAVE to a specified page, or RAND USR (12290+PAGE) to LOAD from a specified page. After loading a program from tape, the DOS must be initialised by the command RAND USR 12865. This command initialises the FDC and loads certain DOS parameters into bytes 16507/16508. Thus, the lines:

```
9900 SAVE "PROGRAM"
9910 CLS
9920 RUN
```

...would be replaced by the lines:

```
9900 RAND USR 12865
9905 RAND USR 12722
9910 CLS
9920 RUN
```

In this example, a GOTO 9900 would save the program to page 2 on the disk, and autorun. When the program is reloaded, it will come up in the same mode (SLOW/FAST) as when it was saved. To save the program on, say, page 6 instead of page 2, line 9905 would read "RAND USR 12726".

Having saved a program, it must be entered in the directory. So, entering RAND USR 13303 brings up the SADOS menu. Pressing E gets a prompt asking which page you want to name, followed by entry of the program name. Having done this, you must re-save the updated directory to page 1. This is offered as a Y/N option when you finish making directory entries, or it can be done by pressing S at the main menu. Now, if you hit L at the menu, the directory appears, listing pages 1 thru 20 and their contents. Any program may be loaded by entering its page number, or you may copy the directory to your printer.

D, for DISK UTILITIES, provides selection of PROGRAM or DATA mode, of Disk Drive A,B,C, or D; Single or Double Density, and also a Clear Directory command. In Data mode, all Variables are saved & loaded, without the BASIC program. When re-loading data, no test is made for variables space. Variables must already have been dimensioned or declared so there is enough room between VARS & E\_LINE for the disk file to load into. With an understanding of these limitations, the Data mode can be very useful. I've successfully programmed a version of WSII.5 which saves & loads data files ONLY; up to 20 16,000 byte text files, each saved with a filename string. The program will sequentially search a disk for any named file, and print an automatically maintained directory. I'm sure that versions of ZX PRO-FILE (or other file handling programs) could be modified to sequentially search a whole disk, if necessary, to find an entry.

CPU and BUS LOADING: I had just gotten a few programs saved to disk, and was marveling at the speed of program access, when I realized I had problems- random load errors galore. Bytes were simply dropping out of programs. Suspecting bus speed/loading problems, I singled out 2 possible culprits: my CPU, and the ribbon cable provided by Aerco. The NEC D780C CPU in my computer is a notoriously substandard Z80 chip, so I replaced it with a new Zilog Z80B. The Z80B is designed for operation at up to 6MHz., thus has lower intrinsic capacitance, and "stiffer" bus drivers than the slower versions. At the same time, I nixed the ribbon cable, and plugged everything into my new JLD 6-slot motherboard. A ribbon cable can have 5-10 times the capacitive crosstalk as a motherboard the same length, so I figured this would help. Not so simple, though. The male edge connector on the FD-ZX is a mirror-image of that needed by the JLD motherboard! Thus, I had to fashion a "side-swapper" adapter to plug everything together. Having done all this, the problems completely disappeared, and the disk system has been 100% reliable ever since. I don't know whether the CPU or the motherboard was the significant factor, but the combination licked the problem.

IN CONCLUSION, the AERCO FD-ZX system has made a "disk convert" out of this ZX81 hobbyist. The hardware is of good quality and is well assembled. The DOS control program is certainly utilitarian, although some resourceful programming can make it do some impressive things. The existing DOS control software simply doesn't take advantage of the FD-ZX hardware's potential. Aerco's documentation is quite thorough, and provides a few clues to accessing more sophisticated DOS commands thru machine code. The incredible speed of reliable access to a large number of programs or files makes it easy to overlook the shortcomings of the DOS.

## HOW TO MAKE BACK-UP COPIES OF SPECTRUM COPY-PROTECTED PROGRAMS

Karim Rahemat

Special thanks to Ariel Frailich for advise and for technical information.

NOTE: THIS ARTICLE IS PROVIDED FOR USE WITH YOUR PERSONAL LIBRARY ONLY AND NOT FOR STEALING COPYRIGHTED SOFTWARE.

How many times have you tried to make a back-up copy of your new software? It seems that whenever you try the copy you have, it doesn't work. Some of these programs may have anti-merge, headerless files, jerky loaders, short loaders, files too close together, fast loaders, false headers; some may be 48k long or have yet other types of copy protection.

Let's discuss some of these copy protection devices. First we have the anti-MERGE program, which prevents the BASIC loader from being MERGED; it can only be LOADED. This can be bypassed with fast reflexes and knowing when to hit the BREAK key. Start by using the header reader program to find the autostart line number of the program. Start LOADING the program and press the BREAK key when the LOAD is just about to finish. (Experiment here.) You should see an error report stating that BREAK was pressed in some line number. Save the program with the original autostart line number.

The next step is to determine whether the next program is headerless, machine code or BASIC. If it's less than 48k long - say 38-40k - LOAD the code at a higher address than normal. This prevents the program from autorunning. If the file doesn't start with a header, then a special LOAD routine is used for LOADING the headerless file. This code may be located anywhere in memory and the most likely place is in a DATA statement for POKEing or in the variables area. Use the header reader on the BASIC program and find the autostart number. LOAD the program as described above for anti-MERGE programs. Now SAVE the file without any autostart line. LOAD HOT-Z and PRINT PEEK 23627+256\*PEEK 23628. This will tell you where the variables area starts. Use HOT-Z again at the address you have found above.

You should now see three LD's, a SC and a call to 0556H. If you have found it, Congratulations! You have just found a routine for LOADING the headerless file. Here's how it works: first the IX register is LOADED with the start address of the file. The DE register is LOADED with the number of bytes, then the A register is LOADED with FFH or 00H - FFH to load the Data portion or 00H to load the Header portion. A CALL is made to the home ROM routine which does the actual loading. Finally, the routine jumps to the the program which has just been loaded.

EXAMPLE: LD IX,8000H - START ADDRESS OF FILE  
LD DE,1800H - NUMBER OF BYTES TO LOAD (Program length)  
LD A,FFH - LOAD HEADER ONLY  
SCF - SET CARRY FLAG  
CALL 0556H - GOTO LD BYTES ROUTINE IN ROM  
JP 8032H - RUN PROGRAM

LOAD routines for headerless files usually consist of only 7 bytes. This also applies to false headers. Fast LOADers are the trickiest of all protection devices. They are very fussy

about clean heads and azimuth alignment. Fast LOADers cannot be fast LOADED themselves; they require a normal (1500 baud) BASIC and machine code program to be loaded first. Every fast LOADER I have hides its fast LOAD code in the Variables area. This can only be discovered with a header reader. The header reader will let you know if any Variables area has been taken. If so, the code is in that area; if not, the code is in either a REM or DATA statement. To find out where the code starts in the Variables area, use PRINT PEEK 23627+256\*PEEK23628. Then use HOT-Z to disassemble from the address you found with the PEEK. The code varies from program to program. You'll just have to experiment. Incidentally, all the above is true for short loaders, jerky loaders and 48k long headerless files. Jerky loaders or pulsed loaders are the most troublesome type of copy protection; copying can only be done via tape to tape. This protection device is now in widespread use. If your program uses jerky or pulsed loaders, you will see the bars (while LOADING the program) go up and down and a short, almost inaudible bleep will go off between bars. As mentioned above, use tape-to-tape copy for back-up. Trying to rid the BASIC of colour control characters is some times difficult: the text bytes themselves may have been altered; the RASP system variable may have been POKEd with 255; and other POKEs may exist. This complicates things a lot. Your best bet is to use HOT-Z to disassemble the BASIC program proper and experiment.

In 1984, a new copy protection device was introduced. It has not been extensively used by software houses yet. It goes by the name of TONING and is accomplished by altering the LOAD header patterns. This too requires special code for LOADING the program into the computer and, likewise, the loader part is LOADED normally. There are three types of tone loaders: wide, short and narrow. The "wide" loader produces bars which are further apart than normal bars. The "short" loader shortens the "silent gap" between programs - this "silent gap" is created and expected by the computer and must therefore be present. This "short" loader is of course not LOADable normally. The "tone" loader uses narrower bars than normal. Use tape-to-tape copy for backing up Toned programs.

Let's now discuss Tape Copiers. We're not talking about the tape copiers which copy programs from one tape to another via the computer (e.g., Cameron Hayne's Tape2Tape). These new ones will LOAD the program into the computer and the SAVE it using the ordinary SAVE routines, giving you clean LOADs. Note that these programs are intended for back-up copies only, not for pirating! Some of these programs are called Kklone 1.0 and 1.1, Pirate7b, Jpc1one, 007spy, Maxspy, Tricky, Tape copier 7, Tape copier6. The tape copier will LOAD the file either headerless or with the header and SAVE it in its original form. Special copiers will LOAD a full 48k program either headerless or with header or a file larger than 40k long. Some will even copy a 61k long file and still work without errors. When backing up a program with a tape copier, don't expect too much. These programs can obviously only handle certain situations; again, the only way to find out is to experiment. Your best bet is to try several programs and use your machine code know-how. Software houses are of course continually developing new techniques in software protection. Some programs require you to purchase a piece of hardware along with the program. The



Cont. from page 6

program checks for the presence of the piece of hardware and then allows you to make backup copies (which of course requires the same hardware to run). And ... if the program doesn't detect the hardware, the program NEUs itself.

---

## QL-CORNER

by: Enore Garconio

I have had the QL now for approximately 3 months and I can safely say that it is a superior home computer, both in price and performance. As advertised, it is also a reasonable computer for a small business. It certainly has enough software out now to run a small business. Compared to the bad old days of a few months ago, there now are at least over thirty programs in the market for it, ranging from games to utilities, languages and business. More and more hardware packages are coming on the market now as well, for example disc drives with added memory, memory boards to increase internal memory to 640K, and I.C.E. (Icon Controlled Environment). It appears that software and hardware companies are believing that the QL is here to stay.

In this article I don't plan to defend the reasons for buying a QL over other computers in the market now (Atari 520ST, Amiga, Commodore 129). I happen to think that the QL bears any computer in its price range. What other computer comes with four very user friendly programs for the total price of \$599.00. If the four packaged programs were being peddled as I.B.M. software (i.e. MS-DOS software) they would cost you several hundred dollars. For example, Wordstar costs over \$500.00 in its MS-DOS format and I believe that Quill is not only much easier to learn and use, but is also as good. As another example, Multiplan (a spreadsheet) costs more than \$250.00, Abacus approaches it in power and beats it in ease of use.

I'm often asked whether the QL is I.B.M. compatible, obviously not, and I do not miss that lack of compatibility in the least.

Neither the computer nor the packaged software has let me down so far. The microdrive cartridges are sometimes poorly made but the microdrives are perfectly fine. The only real weakness then of the QL are the microdrive cartridges and possibly the slow speed of its basic (it is not really that bad). I solved both problems by buying the Technology Research Delta Disc Drive Interface. It comes with an extra 128K of RAM and a centronics parallel port (all for the price of \$279.00 U.S. from Curry Computers). This interface gives me more data security and a satisfying increase in speed both in data transfer and, for some funny reason, in its basic. I will report my impressions of it in a later article.

There are a couple of programs that you should know about, both are very useful. "The Cartridge Doctor" is an absolute requirement. It can actually recover files from corrupted tapes. A couple of times I was able to recover Archive files which kept insisting that they were the wrong file types. Both times I used the file patch utility of the Doctor to change a missing byte in the header. The Doctor will also back up your tapes as it checks them out. It is well worth the \$21.95 U.S. The other is the QL Toolkit. If you do any programming with the QL it's a must. It has full screen editor, a spooler, a real time clock that appears in window#0, three ways of packing up a tape, multitasking, random file capability, etc. etc. etc. There are too many programs to list in this article. It is also well worth the price of \$26.95 U.S.

There is an attempt to start a QL part to the Timex/Sinclair Club. If there is any interest out there let George Chambers know and we can get it going. If you know someone who has a QL you might point out to him that he is not alone. We have begun a QL folder in the Club's library, we need contributions in articles and old books. If you are willing to copy some of your programs we can get a program library going. If you've copied any programs from magazines you might think about donating them to the club. It would also be nice if any of you can contribute your thoughts on the QL for the Club's newsletter. The more people see how good the computer is and the more the QL part of the Toronto Timex/Sinclair Club will grow. Anything is worth it right now since we are at the very beginning.

Up the QL.

### A METHOD TO PROTECT MORE THAN ONE LINE FROM DELETION (ZX81)

First, before entering anything else, enter the line or lines to be protected.

Next enter POKE 16509,40 (Actually you could POKE any number between 40 and 63). This should change the number of the first line to a letter followed by 3 digits.

Then enter the rest of your program

The lines present when you POKE'd 16509 will then be visible at the end of the program but cannot be changed or deleted in the normal way, and now there is no easy way of finding out which address has to be POKE'd to change the line number back to a "proper" one, as entering the main body of your program will have moved the POKE'd byte up in memory.

# BOB'S NOTEBOOK

```

=====

```

## BINARY SEARCH SUBROUTINE

This program is based on one by John Gilbert published in Timex Sinclair User Vol.1 #4.

Compared to the slower serial search often used in BASIC programs, the binary method is quite fast, doing its job in about one tenth of the time. The reason is that while the serial method looks at every record in a file, the binary does not.

First, all records must be sorted alphabetically (or numerically) for the binary method to work; meaning sorted according to the field to be searched on.

Here is how it works: (numbers in brackets refer to line #s)

1. A search data string (s\$) is entered. This may be a word or words or part of a word or a number. [2220]
2. The start of the field being searched is placed in variable ss. [2230]
3. Variable s takes on the value of the half way point. [2340]
4. With the pass counter (p)=0 the record at the half way point is examined for a match and if found, then the record is printed [2400]; the pass counter incremented [2420]; variable s1 used to store the initial value of s [2430] and s incremented [2480].
5. If a match is not found then the record is examined to see if it is higher or lower than a\$. If lower, the record will be taken as the end of the file and the other half cut away [2380]; if higher, the record will be the start of the file [2370]. We now have a new value for s which is in the middle of the already halved list.
6. The search continues upward (alphabetically) until s>bb when the search reverses direction downward ending when s1=ss [2500]; printing the record in the downward direction takes place [2510] and s1 is decremented [2530].
7. When records are printed, they will not be in strict alphabetical order; this could be corrected at the expense of time and memory.

## VARIABLES USED

aa=start search  
bb=end search (in set by user)  
p =pass counter  
s1=temporary storage for s  
a\$=search data string  
s =search limit half-way point  
r\$=data file being searched  
(may be changed throughout to suit user's needs)  
n =number of records being searched (set by user or user program)  
ss=start of field used for search; now set at 1 but can be changed to suit user needs  
pz=limits search if no record found.  
N.B. If any of these variables clash with those in your program, change them accordingly. Also, you may wish to renumber this program to suit your needs.

Bob Mitchell  
October 1985

## PROGRAM LISTING

```

2200 REM SEARCH
2210 LET pz=0: LET aa=1: LET bb=
n+1: LET p=0: LET s1=1
2220 INPUT ("enter search data")
' LINE a$
2230 LET ss=1
2240 CLS
2250 PRINT #1;AT 0,0;"Searching
for ";a$
2340 LET s=INT ((bb-aa)/2+.5)
2350 GO TO 2390
2360 IF s>bb THEN GO TO 2540
2370 IF r$(s,ss TO ss+LEN a$-1)
a$ THEN LET s=INT ((bb-aa)/2+.5)
2380 IF r$(s,ss TO ss+LEN a$-1)
a$ THEN LET s=INT ((bb-aa)/2+.5)
+aa
2385 IF p=0 AND pz=10 THEN CLS :
KEEP 1,-20: PRINT AT 10,0;"NO R
ECORD FOUND WITH" a$: PAUSE 120:
RETURN
2390 IF p>0 AND r$(s1)=r$(s) THE
N GO TO 2500
2400 IF r$(s,ss TO ss+LEN a$-1)=
a$ THEN PRINT r$(s)
2420 IF r$(s,ss TO ss+LEN a$-1)=
a$ THEN LET p=p+1
2430 IF p=1 AND r$(s,ss TO ss+LE
N a$-1)=a$ THEN LET s1=s
2440 IF r$(s,ss TO ss+LEN a$-1)=
a$ THEN GO TO 2480
2450 IF r$(s,ss TO ss+LEN a$-1)
a$ THEN LET bb=s
2460 IF r$(s,ss TO ss+LEN a$-1)
a$ THEN LET aa=s
2470 LET pz=pz+1: GO TO 2360
2480 LET s=s+1
2490 GO TO 2360
2500 IF s1<aa THEN GO TO 2550
2510 IF r$(s1-1,ss TO ss+LEN a$-
1)=a$ THEN PRINT r$(s1-1)
2530 LET s1=s1-1
2540 GO TO 2500
2550 RETURN

```



# LUNAR CALCULATING

by Mel Richardson

The readily accessible T/S 1000/ZX81 graphics are not about to impress anyone but the Magic Membrane is no slouch when it comes to it's floating point arithmetic. The program that follows is a good example of the ZX's ability to handle complex calculations with great accuracy.

A table of dates and times of the NEW and FULL phases of the moon can be generated for any year from several thousand years in the past or the future with accuracy of + or - 2 minutes at the extreme dates. Lines 150 and 160 cause the Julian calendar to be used for years 1582 and earlier and line 240 corrects for non uniform-rotation of the earth by converting from Ephemeris to Universal time.

```

10 REM NEW AND FULL MOON AD
APTED FROM SKY AND TELESCOPE, HA
R 85
20 CLS
30 SLOW
30 PRINT TAB 5;"NEW AND FULL"
NEW AND FULL
40 PRINT
50 LET U=0
60 LET F=0
70 LET R1=P1/180
80 LET U=0
90 PRINT "ENTER YEAR: (YYYY)"
100 INPUT Y
110 PRINT AT 2,0;" "
120 FAST
130 PRINT AT 2,9;"*** ";Y;" ***"
140 PRINT
150 LET G=1
160 IF Y<1583 THEN LET G=0
170 LET K0=INT ((Y-1900)*12.368
5)
180 LET T=(Y-1899.5)/100
190 LET T2=T*T
200 LET T3=T*T*T
210 LET J0=2415020+29*K0
220 LET F0=0.0001178*T2-0.00000
0155*T3
230 LET F0=F0+0.75933+0.5305886
8*K0
240 LET F0=F0-0.000837*T-0.0003
35*T2
250 LET J=J+INT (F)
260 LET F=F-INT (F)
270 LET M0=K0+0.03084821133
280 LET M0=360*(M0-INT (M0))+35
0.2242
290 LET M0=M0-0.0000333*T2
300 LET M0=M0-0.00000347*T3
310 LET M1=K0+0.07171366128

```

```

320 LET M1=360*(M1-INT (M1))+30
0.0253
330 LET M1=M1+0.0107306*T2
340 LET M1=M1+0.00001236*T3
350 LET B1=K0+0.08519585128
360 LET B1=360*(B1-INT (B1))+21
.2964
370 LET B1=B1-0.0016528*T2
380 LET B1=B1-0.00000233*T3
390 FOR X=0 TO 28
400 LET U=U+14*X
410 LET F=F+0.765294*X
420 LET K=X/2
430 LET M5=(M0+K+29.10535608)*R
1
440 LET M5=(M1+K+365.81591808)*
R1
450 LET B5=(B1+K+390.67050548)*
R1
460 LET F=F-0.4088*8IN (M5)
470 LET F=F+(0.1734-0.000393*T)
*8IN (M5)
480 LET F=F+0.0161*8IN (2*M5)
490 LET F=F+0.0104*8IN (2*B5)
500 LET F=F-0.0074*8IN (M5-M5)
510 LET F=F-0.0051*8IN (M5+M5)
520 LET F=F+0.0021*8IN (2*M5)
530 LET F=F+0.001*8IN (2*B5-M5)
540 LET F=F+.5/1440
550 LET J=J+INT (F)
560 LET F=F-INT (F)
570 IF U=0 THEN PRINT "NEW"
580 IF U=1 THEN PRINT "FULL"
590 GOSUB 6600
600 LET U=U+1
610 IF U=2 THEN LET U=0
620 NEXT X
630 LET F=F+.5
640 IF F<1 THEN GOTO 0670
650 LET F=F-1
660 LET J=J+1
670 IF G=1 THEN GOTO 0700
680 LET A=J
690 GOTO 0720
700 LET A1=INT ((J/36524.25)-51
.12264)
710 LET A=J+1+A1-INT (A1/4)
720 LET B=A+1524
730 LET C=INT ((B/365.25)-.0345
)
740 LET D=INT (365.25+C)
750 LET E=INT (16-D/30.61)
760 LET D=D+INT (30.61+E)*F
770 LET M=E-1
780 LET Y=0-4740
790 IF E>10.5 THEN LET M=M-12
800 IF M<2.5 THEN LET Y=Y+1
810 LET D1=INT (D)
820 LET H=24*(D-D1)
830 LET H1=INT (H)
840 LET M9=INT (60*(H-H1))
850 PRINT TAB 8;Y;TAB 13;M;TAB
18;D1)
860 PRINT TAB 22;H1;TAB 24;"H";
TAB 26;M9;TAB 28;"M"
865 IF X=28 THEN STOP
870 RETURN

```

## TO CUT DOWN ON LOAD AND SAVE TIME

To cut down on the amount of time required to save and to load programs, eliminate all the variables first, before SAVE'ing. This can be done by using a CLEAR command before SAVE'ing.

Of course this technique should not be used if there are variables in the program which must be saved.

## PLOT, CLS and SCROLL on the TS2068

Cameron Hayne  
11 Pusey Lane  
OXFORD OX1 2L8  
U.K.

In a previous article, I discussed how to use the ROM routines for printing from machine code. In this article I will explain a bit more about some of the other display routines in the ROM. Some of this will have easy application from BASIC and so I will give addresses in decimal as well as in hexadecimal. The corresponding addresses for the SPECTRUM will be given in parentheses.

To PLOT a point on the screen is very easy from machine code and vastly faster than from basic. First you must load the BC register pair with the pixel coordinates of the point to be plotted: B=y coordinate, C=x coordinate. Then you call the ROM routine and voila! The ROM routine is at hex 263E, decimal 9790 (hex 22E5, decimal 8933). For example:

```
LD BC,800FF (the # indicates hex)
CALL #263E will plot a point at the right hand bottom of screen
```

The ROM routine uses the "temporary" attributes when plotting a point and these can be set using RST #10 as I explained in the previous article. Note that the BASIC PLOT command automatically sets the attributes to FLASH 8 ; BRIGHT 8 ; PAPER 8 and it may be necessary to do this when PLOTTING from machine code in order to not disturb what is already on the screen.

As you know from BASIC, the screen is split into two parts : the main screen (rows 0-21), and the lower screen where the reports are printed. The size of the lower screen is determined by the system variable DF\_SZ at address hex 5C6B, decimal 23659. (Since IY is supposed to be hex 5C3A, this can also be referred to as IY+49 decimal or IY+31 hex.) Usually (DF\_SZ)=2 but the lower screen can be expanded to more than two rows or contracted to zero by poking DF\_SZ. (But make sure it is at least 2 before returning to BASIC!) The attributes for the lower screen are contained in BORDCR. You can print on the lower screen from BASIC using PRINT #1; etc. and the AT coordinates are taken as relative to the top of the lower screen. (Thus 0,0 is the first position in the lower screen.) When you do this, the lower screen expands as necessary. The same can be done from machine code by first changing the stream with: LD A, 01 ; CALL #1230 (SPECTRUM #1601). Whereas the whole screen can be cleared by CALL #08A6, decimal 2214 (#0D6B, decimal 3435), the lower screen alone can be cleared by CALL #08A9, decimal 2217 (#0D6E, decimal 3438). On the other hand, if you wanted to clear a certain number of rows at the bottom of the screen (disregarding the boundary between the lower and main screens), you could do this by loading the B register with that number and calling the ROM routine at hex 097F, decimal 2431 (hex 0E44, decimal 3652). For example: LD B, #0C ; CALL #097F would clear the bottom half of the screen.

My final topic is scrolling. The BASIC PRINT routines are the RST #10 routine will issue a "Scroll?" query when they have reached the end of the main screen (when (DF\_SZ)=2 as usual).

Automatic scrolling of the whole screen can be achieved by poking the system variable SCR\_CT with a number bigger than one. To scroll a certain number of rows at the bottom of the screen (disregarding the boundary between the lower and upper screens) up one row, you load the B register with the number of rows, then call the ROM routine at hex 039B, decimal 2363 (hex 0E00, decimal 3584). For example: LD B, #0B ; CALL #039B would scroll the bottom half of the screen up one row.

Here is a short demonstration program;

```
1 GOTO 100
10 DEF FN H(x)= INT(x/256): DEF FN L(x)=X-256*FN H(x)
20 RESTORE
30 READ byte : IF byte (>) PI THEN POKE address, byte : LET
  address = address + 1 : GOTO 30
40 RETURN
50 DATA 6, n, 205, FN L (romaddr), FN H (romaddr), 201, PI
100 LET address = 60000 : LET n = 12
110 LET clsbottom = address : LET romaddr = 2431 : GOSUB 20
120 LET scroll = address : LET romaddr = 2363 : LET n=n-1
  : GOSUB 20
200 FOR r = 0 TO 21 : PRINT "row ";r: NEXT r
210 FOR r = 0 TO 1 : PRINT # 1; AT r, 0; "lower row ";r:
  NEXT r
220 PAUSE 200
230 PRINT AT 12, 10; "POOF !" : PAUSE 20: RANDOMIZE USR
  clsbottom
240 PRINT #1; AT 1, 0; RND
250 PAUSE 20 : IF USR scroll THEN GOTO 240
```

Now that I am in England and my public library has about 100 books on the Spectrum (but 90 % out on loan) I am discovering some good sources of information. Most people will already know of "The Complete Spectrum ROM Disassembly" by Drs. Ian Logan and Frank O'Hara (Melbourne House, 1983). Their annotations are excellent and the book is invaluable to me. A new discovery is "The Spectrum Operating System" by Steve Kramer (MicroPress, 1984). He explains how to use many of the ROM routines, including the floating point calculator, the microdrive routines and how to use the Interrupts. I thought it worth the \$6.00 asking price and now own a copy. An excellent general reference book on the Z80 machine code is "Z80 Assembly Language Subroutines" by Leventhal & Saville (Osbourne/McGraw-Hill, 1983)

TO PLACE A PARTICULAR LINE NUMBER AT THE TOP OF  
AUTOMATIC LISTINGS (ZX81)

First move the cursor to a line number greater  
than the one you want at the top. Then enter:

```
POKE 16419, N-INT(N/256)*256
POKE 16420, INT(N/256)
```

Now, when you press NEWLINE the automatic  
listing will begin where you specified (N is the  
number you want at the top of the screen)



## CLOCKING THE 2068

Our 2068's have a built-in clock, sometimes called the Frame counter. This clock starts at zero every time the computer is turned on. It increments the value at the variable address 23672 every 1/60th of a second until it reaches 255. This takes about 4 1/4 seconds and then resets to zero and increments the value at 23673. This counter will also go up to 255 but only every 18 minutes. At that time it, in turn, increments the final counter at address 23674. The final counter reaches 255 after about 3 days and 6 hours.

The counters can be reset to zero or any other value by POKEing the number into three addresses.

The accuracy of the clock is .01 percent (+/- 10 seconds per day). The clock counts up as long as the 2068 is left on, with three exceptions:

- During a BEEP,
- During tape operations (SAVEing and LOADing),
- During printing on the printer.

Under the above three conditions, time will stand still and continue when the condition is completed. The following expression gives the present count on the clock:

```
(65536 * PEEK 23674 + 256 * PEEK 23673
+ PEEK 23672) / 60
```

When using this expression in a program it is possible to PEEK 23673 just as 23672 resets to zero. This will give a false reading of 23673 because it should have been incremented at that moment, between reading the two addresses. To overcome this situation, use the above expression twice successively, and take the highest reading.

from : SPECTRUM Users manual, Chapter 18  
compiled by Fred Schakel  
London T/S Users Group

---

SUBROUTINE TO CLEAR A PORTION OF A SCREEN  
WHEN DESIRED (ZX81 BASIC)

```
100 GO SUB 3000
120 PRINT AT 15,9;"YOU HAVE CLEARED THIS PART OF
THE SCREEN"
3000 FOR N = 13 TO 21
3010 PRINT AT N,0;" (32 spaces) "
3020 NEXT N
3030 RETURN
```

## BASICALLY "IN"

An advantage of using the IN function instead of INKEY\$ is that multiple keys can be sensed. If the keyboard was to be divided into 8 groups so that each group was half a row containing 5 keys, then these groups would be represented by IN addresses as follows:

```
IN 65278 = keys CAPSHIFT to V
IN 65022 = keys A to G
IN 64510 = keys Q to T
IN 63486 = keys 1 to 5
IN 61438 = keys 0 to 6
IN 57342 = keys P to Y
IN 49150 = keys ENTER to H
IN 32766 = keys SPACE to B
```

When IN 65278 equals 31 (binary 11111), none of the keys from CAPSHIFT to V are being pressed. If one of these keys was pressed, it would change the binary 1 to a zero. The least significant digit represents the key closest to the outside of the keyboard. For example, IN 61438 is 19 (Binary 10011) indicating that both keys 7 and 8 are pressed. Check this by RUNNING this one-line program: (press keys 6 to 0)

```
10 PRINT AT 1,1: IN 61438: GOTO 10
```

This only uses the 5 least significant digits of the address. The 6th least significant digit is the bit which presently appears at the EAR socket. IN 251 handles the data for the printer as does OUT 251. OUT 254 colours our border on the 3 least significant bits while it's 4th and 5th bits are used for the MIC socket and the BEEPer respectively.

from: SPECTRUM Users Manual, Chapter 23  
compiled by: Fred Schakel  
London T/S Users Group.

---

TO ELIMINATE SCREEN BLINK  
RESULTING FROM PAUSE COMMAND (ZX81)

The solution is to replace the PAUSE command with a FOR/NEXT loop:

```
FOR F = 1 TO Z * 10
NEXT F
```

where Z is a number of your choice that determines the length of the pause interval.

or  
FOR F = 1 TO 600  
NEXT F

A loop of 600 corresponds approximately to ten seconds.



Disk Droppings  
by Greg Lloyd  
Compuserve ID  
76317,1461

To those of you who are living under the impression that loading and saving programs and data to tape is state of the art, listen up. Some TS users have discovered Disk drives. There are a great number of them on the market varying in price and features.

During the next few issues, in this column I hope to cover the features and utilities of all the available disk systems on the market for the TS 2068 and ZX-81 TS 1000.

We'll start by defining some of the main components of a disk system. First you need a computer, a disk drive, a disk controller or interface, a power supply for the disk and the interface and some blank disks.

Assuming your computer works most of the time, the next item is the disk controller. This device does just that, it connects your computer to the drive and translates your commands into a light and sound show on the disk drive. The controller also dictates the number and type of drives you can run on your system.

The drive, regardless of size and type, will store your programs and data in a rapid and reliable fashion. Look for the most widely available format so pricing is competitive and repairs are easily obtainable. Disk drives do not bounce very well!

The power supply +5 +12 volt variety will either come with the drive if you buy a packaged system or you will have to scrounge one. Be sure the output is correct before you try it.

Some things to think consider before you choose a Disk system are:

1 How much memory does it use? In most cases the disk system will use some of your precious ram. Less is best here because you can't add it easily.

2 Will your printer and modem work with it? Some systems will give your printer interface, modem and you fits due to software or hardware incompatibility.

3 Can you run those 9999 Spectrum games on the system? (ZX81 user's can disregard this). This is the most important consideration by far. What's the use of a disk drive if you can't load those games fast!

4 P R I C E. Some systems cost a lot of smackers and some don't. The bottom line is you don't always get your money's worth.

I have to admit I am biased, I own a LarkEn disk system for my 2068 which is the winner in the low price range. The interface (\$120), a power supply (\$30), and a drive (\$165) for \$315 CDN. I can compare it and the other systems to the IBM PC I use at the office. The IBM, incidentally is no speed ball in any category except reducing one's bank balance!

Now you know what's needed and what to look for, next issue We'll see what's out there and what to do with it. 01/14/86

Page 12.

Postmaster, if Undelivered Return to :

Toronto Timex - Sinclair Users Club  
P. O. Box 7274 Stn. A  
Toronto, Ont., M5W 1X9  
Canada

